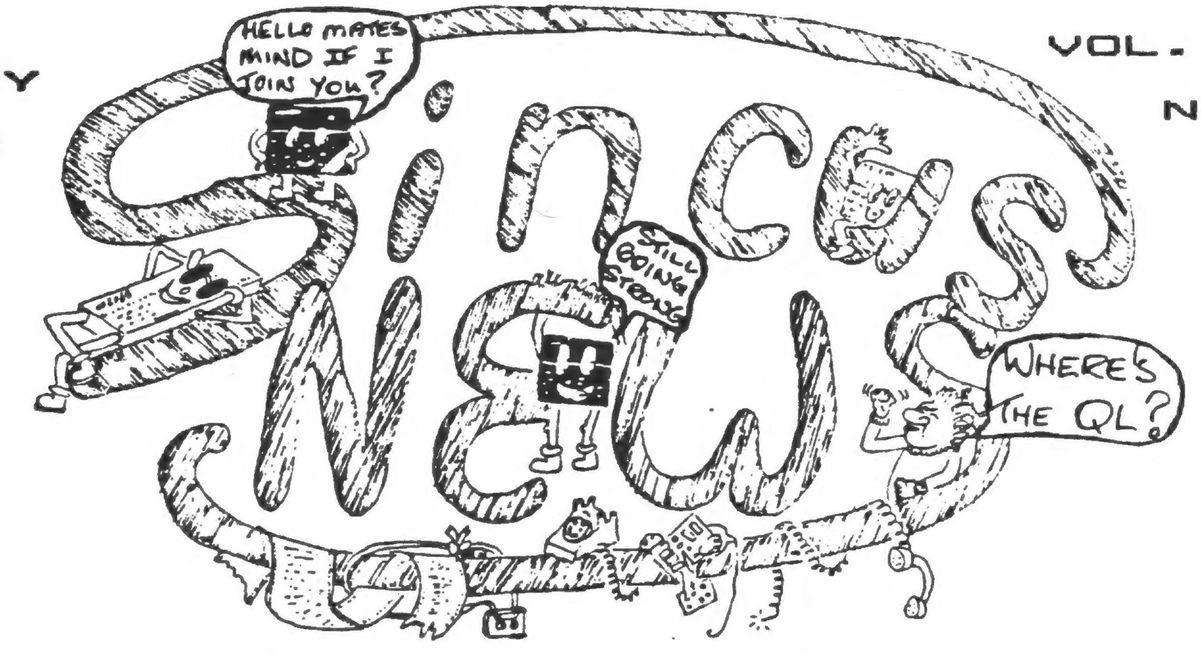
JANUARY
1985

VOL. III
No. 3



+ + WHAT'S INSIDE + +

* * SINCUS ACTIVITIES * *

-CLASSES-	-MEETINGS-	-HARD/ZX81-
[Programming, etc.]	*[Main Thing!!]*	*[Hardware+TS1000]*
TUESDAY January 15 7:00 PM Vestal Library	WEDNESDAY January 16 7:00 PM Chase/First Bank	SATURDAY January 26 10:00 AM Vestal Library
THURSDAY February 28 7:00 PM Vestal Library	WEDNESDAY February 20 7:00 PM Chase/First Bank	SUNDAY February 24 1:00 PM Vestal Library
THURSDAY March 28 7:00 PM Vestal Library	WEDNESDAY March 20 7:00 PM Chase/First Bank	SATURDAY March 23 1:00 PM Vestal Library

The Executive Committee of SINCUS meets the first Wednesday of the month at the Vestal Public Library at 6:30 PM unless otherwise stated in advance. Any regular SINCUS member is welcome!

SINCUS NEWS P.O. BOX 523 OWEGO, NEW YORK 13827

SECRETARY'S REPORT

Wednesday, December 19, 1984, SINCUS meet at the Chase/Lincoln Branch on the Vestal Parkway corner Murray Hill Road, Vestal, NY at 7pm. 23 members attending.

A proposed change in dues and number of newsletters for corresponding members was passed. The dues go from $6/10 issues to $8/12 issues. Postage increases plus we are producing a monthly letter instead of 10 a year necessitates the increase.

Bulk tape sales continues to go well, as well as advertising for the newsletter.

FOR YOUR INFORMATION

We started a consumer hot line for our members this past spring and have dealt with one problem, and settled it agreeably to all concerned. Lately we have been getting reports on one business -Phoenix- of Dover, Del. and while no one has lost money, the complaints are of slow delivery and rude and poor service. We don't wish problems on any one, but as the number of complaints grow so does our concern.

AGAIN OUR SNOW EMERGENCY PROCEDURE

If the Broome County Sheriff declares "Emergency Travel Only" our meet will be cancelled that night...however with the number of meetings planned over the next three and 1/2 months, I doubt anybody will miss much.

CARL TERRY SERVICE AWARD

The first *ANNUAL CARL TERRY SERVICE AWARD* was voted by the society's officers to be given to that member, not an officer, who makes exceptional contributions throughout the year. This first award, a plaque and a year's membership, was given to Wes Brzozowski. Wes, from almost day one of SINCUS, has given talks, written articles and conducted classes. Much thanks Wes, this is long over due.

SINCUS BUYS STUFF and DOES STUFF

A proposal to purchase the Buyers Guide for $20 from D. Lipinski Software,2737 Susquehanna Rd., Roslyn, PA 19001 by the society was made and passed. This catalog of current retail outlets in the US is supposed to be updated and will be at meetings for your information.

A proposal to purchase a Spectrum ROM for the society was made and passed. The ROM will be used by Wes Brzozowski to test feasiblity of a design for an upcoming society project.

We will hold classes in basic soldering and assembly of PC boards in January. See schedule on cover for HARDWARE/1000. is Dale who has given such instruction at his employment has offered to lead the group.

The member users of 81s, 1000s, 1500s have not been forgotten The meetings set up at the Vestal Library for HARDWARE/1000 will be used to get the 1000 users working on their own projects and classes and hopefully articles for the newsletter. There are a lot of resources within the society for 1000 users.

For those interested in Machine Code, write

ZILOG,INC.
ATTN: TECH PUBLICATION Z80 CPU
1315 Dell Avenue
Campbell, CA 95008

Ask for "*Programmer's Reference Guide,Z80 CPU*"
publication # 03-0012-02

Wes demoed "THE HOBBIT", an adventure game by which all other British adventure games are measured. I only saw a little and it is worth getting a ROM to see this one! Wes talked a little on the LOAD procedure, "most headers on commercial programs have the normal 1500 or so cycles of which your 2068 needs only 256, right after the 256 cycles comes a little spike with the info which tells what follows, a program or bytes. The British in an attempt to foil copying have left off the header. The info for the computer comes at the end of the tape loading."...to use British programs you'll need a Spectrum ROM-about $20 or a Spectrum Emulator ($60) or the ROM Switch ($55). Each has it's advantage as previously covered in earlier SINCUS NEWS."

Dave Schoenwetter gave an update on the local BBS situation, another clown is erasing all messages on the local BBS, apparently another jerk is mad because the clown is using his handle. This has gotta be a real selling point on modems!

If you're buried up to your cursor in snow over the next couple months, come on down to the classes, and the hardware projects, learn a little, meet some good folks and have a good time. See you there.

Paul Hill
Rec Scty
SINCUS

Also I've been looking for a good short and sweet input routine to filter out the wrong inputs,so far I like this:

```
100 PRINT "Enter 1 or 2"
110 LET a$=INKEY$: LET a= CODE a$
120 IF a <>49 AND a <> 50 THEN GO TO 110
130 PRINT a$
```

I like it for menu handling and short of hitting the BREAK key you don't stop the program, with wrong inputs. Paul Hill

NEWSLETTER EXCHANGES

This is our current list of UG swappers, we are always interested in hearing from new folks.

T/S Users Group, PO BOX 7274, Station A
Toronto,ONT MSW 1X9 CANADA

LIST, PO Box 438, Centerport, NY 11721

CATS UG. PO BOX 725, Bladensburg,
Maryland 20710

T/S UG of Las Vegas, 2405 Howard Dr.
Las Vegas, NV 89104

CCATS UG.1419 1/2 7th St. Oregon City,
Oregon 97045

Triangle UG 206 James St. CARRBORO,
North Carolina 27510

SLUG, 9800 Mary Dell Ln, Louisville,
Kentucky 40291

South Bay TSUG, PO Box 4133, Santa Clara
California 95054

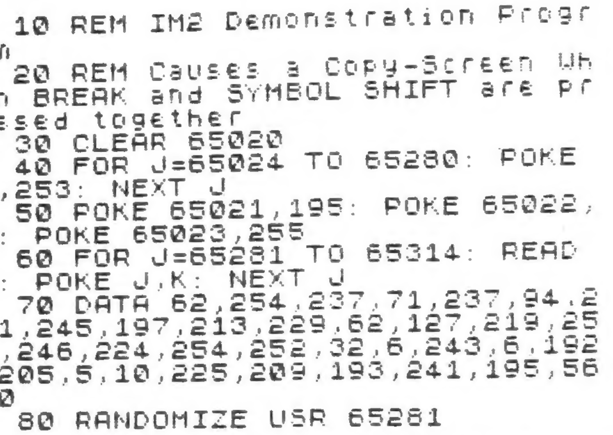
S/TUG of BCS, 284 Great Rd, APt D-5,
Acton, MA 01720

T/SUG of Cincinnati, 11 Funston Ln,
Cincinnati, OH 45216

John Kuhn, 1707 King St. JAcksonville,
Florida 32204

We have mailed out over the past 2 months over 25 of our newsletters in trying to expand our swap network (one response to date) -either the rest are no longer publishing or didn't get our newsletter for one reason or another.

from "COMPUTUS INTERRUPTUS" pg 7-8



```
  10 REM IM2 Demonstration Progr
am
  20 REM Causes a Copy-Screen Wh
en BREAK and SYMBOL SHIFT are pr
essed together
  30 CLEAR 65020
  40 FOR J=65024 TO 65280: POKE
J,253: NEXT J
  50 POKE 65021,195: POKE 65022,
8: POKE 65023,255
  60 FOR J=65281 TO 65314: READ
K: POKE J,K: NEXT J
  70 DATA 62,254,237,71,237,94,2
01,245,197,213,229,62,127,219,25
4,246,224,254,252,32,6,243,6,192
,205,5,10,225,209,193,241,195,56
,0
  80 RANDOMIZE USR 65281
```

TYPE IN - RUN - then LOAD YOUR FAVORITE PROGRAM SYM. SH+ BREAK = "COPY"

The following are "clippings" from other newsletters th this info we gather be made known to all our readers:

From CATS: Sinclair in England has released the Spectrum 4. It is enclosed in a QL size keyboard....

Next from SUM Magazine Problems with TS 2050 modem and power strip?? "We hear from our customers that they have a problem loading the software tape that comes with the Westridge modem. If you are using a power strip move your plug on the recorder to another outlet and it should work. Does anyone want to comment on that??" (PH note:60Hz hum??).

FromTIMELINEZ (SBTSUG) from TS Users of Cincinnati, by Gary Szekeres: The BUG in Vu Calc(2068) LINE 3200 should be:

3200 CLEAR 29327:DIM b$(100):DIM c$(20): etc.

rest of statement intact. The CLEAR 29327 is missing and cause the clear work sheet routine not to work. to get back into VuCalc GOTO 3200(ENTER). ...from Charles R. Byler, Cpt US Army, HQ, USAREUR, & 7th Army, Box 1882,APO NY 09063..."TS1000/ZX81 owners to load and stop auto run machine code program, enter as a dir- ect command FAST, then RAND USR 836. Play the program. Program will load and stop with a report code C/0. ..from CATS the area in RAM used for Bank switching starts at 25365 and contains 323 bytes. This space can be used for MC so long as bank switching isn't being used. Unused fixed addresses are located in the system variable area at 23681, 23728-9 and 23747. Thanks to Bruce Nickel of CATS.

FROM SINCUS thanks Stan Livingston

FOR THE ZX81/TS1000/1500

```
5   REM HEX > DEC
10  POKE 30000,1
20  POKE 30003,201
30  LET X= 30001
40  LET A$=""
50  IF A$=""THEN INFUT A$
60  PRINT A$;
70  IF A$ =" "THEN GOTO 120
80  POKE X,16*CODE A$+CODE A$(2)-476
90  LET X=X+1
100 LET A$=A$(3 TO )
110 GOTO 50
120 PRINT "  ";PEEK (X-2)*256+PEEK(X-1)
```

Run it and ENTER a hex number "FFFF" and it will print 65535

Jim Webb, our newest SINCUS member gave this tip on the ZX81 TS1000 machine. To clear up TV reception, Jim drilled small holes in the plastic body and improved the grounding by installing small metal wood screws, wrap copper wire from ground strap to screw head, also did likewise to the 16K RAM pac.

Anyone have problems getting "INVALID COLOR" when you aren't even using color? Happens when I use the INVERSE VIDEO every now and then haven't been able to make it happen. Anyone else? I've gotten a flashing letter in the listings be SHIFT GRAPHICS in extended mode once and cannot cause it to happen again!!!

Paul Hill
SINCUS

TS1000/ZX-81 COMMENTS

from Jesse Peeler, Costa Rica Users Group

TECHNICAL COMMENTS RELATING TO THE SINCLAIR ZX-81 AND THE TS1000 COMPUTERS AND EXTERNAL 16K RAMs

P.C. Boards are identified as issue 1 or issue 3. Both versions are found in ZX81s/only issue 3 is used in the TS1000. Electrically, they are almost identical. Issue 3 is an improved layout with a neater appearance. ZX81s have ICs mounted in sockets. The TS versions have their RAM chips hard soldered in place. All ZX81s were built in England or sold in kit form. Most TS were built in Portugal, but some were built in France. Quality control was highest on ZX81s. Poor quality control was observed on TS units, particularly those manufactured in Portugal. Fabrication defects most observed were faulty installation of the flexible PC "fingers" from the keyboard into the special sockets. This defect has been observed in almost 100% of TS units manufactured in Portugal. Unsoldered/partially soldered ground busses were observed in both Portuguese and French units.

ICs(Integrated Circuits)

All units use a SCL (Sinclair Computer Logic) special purpose chip manufactured by Ferranti-ULA 2C184E. (Ferranti will not even acknowledge letters requesting info or cost).

All units use a D2364C ROM. CPUs vary. ZX81s have NEC P1Z308-151 or D780c-1. TS uses Zilog chips. All are Z80A chips in one form or another.

RAMs, INTERNAL

Most variation is seen with the RAM. In the ZX81, only 1K of RAM is built in. The PC board was cleverly designed to use either 2 @ 2114s or a single 4118. The 2114 is a 1K x 4 static RAM, whereas the 4118 is a 1K x 8 dynamic RAM. TS internal RAMs are all 2K in various versions from different manufactors, i.e., Toshiba 2016P-1, Motorola 2CM38818C, NEC D4016D-1,Toshiba TMM 2016P-1.

TRANSISTORS

ZTX-313s are normally used I have found uMPS-2369s mounted in one computer. Also, I have found that an MPS-3563 works well as a substitute.

LOAD/SAVE Modifications

Change R-27 to 27ohms and C-11 to 0.015 mfd.

RAMs, EXTERNAL (16K variety)

PC boards are identified as issue 1, 2, or 3. Issue 1 and 2 are composed of 2 small PC boards that are folded inside the case. The differences between 1 and 2 are minimal-an additional diode and decoupling capacitor. Issue 3 combines many logic functions in a single Ferranti ULA1H035E chip to reduce the chip count by 5 and a single PC board is used, mounted in the same case. (Ferranti won't answer any questions on this chip either).

Transistor is either a ZTX-750 or ZTX-752. Recent French and Portuguese units used either a 2N6727 or MPS6727.

Wobble is often a problem-varies with units. The French units utilize a PC connector that has the tightest fit resulting in least wobble of all units observed. (Unfortunately, ZX80 and ZX81 were chosen as names of Sir Clyde's(sic) first two computers. There is a tendency to confuse these designations with Z-80 and Z-80A, which are the CPUs. Therefore, remember that Z-80A is a Zilog designator for their 4MHz CPU.)

POWER SUPPLY PROBLEMS

Once in a great while you get a noisy power supply. In such cases,the bridge rectifier is first suspect. (A power supply can still partially function with 1 or 2 diodes bad- but it will be noisy!) You must crack open the power supply case and find the faulty diode(s) and replace them with 1N002 diodes. I've never seen a capacitor fail, but it could and the replacement is a 1000 mfd /16V capacitor.

I recommend cracking open the power supply whether there is a problem or not. I can put a miniature SPST switch in series with the output so that I can kill power at the power supply, rather than pulling the plug at the computer.

To avoid drop-outs due to looseness of the power supply plug, I removed the power jack completely. (Desolder it and remove.) I then hard wire the power wires in place, tack them securely with some silicone rubber and the power drop-out problem is completely solved. For an even neater job, one should consider putting a small male and female coonector near the power supply to disconnect the system. Watch out you don't reverse polarity!

HEAT AND THE ZX-81/TS1000

Where I live heat is not a problem. However based on the vast amount of letters and complaints seen in SYNC and SYNTAX, heat is a problem for many and the logical solution- the only one I've not seen presented -is to get the primary heat generator outside of the computer case, like so:

1. The primary heat generator is the 7805 3-terminal 5 volt regulator. The higher the input voltage applied to the this device the more energy which must be dissipated in heat by this device to reduce the output voltage to +5VDC.

2. Input voltage (from external power supply) varies from 11+ volts down to 7 volts. Variation comes primarily from add-ons which are connected to the computer. Each device added pulls more current- which causes voltage to drop. (Should you add too many external items, say with a 750ma power supply, the response would be too high a current drain, voltage would drop too low and the computer would quit functioning.)

3. Desolder the 7805 regulator and remove it along with the aluminum heat sink. Now, combining with (step 3 above) connect a 3-wire input to the computer. Three wires are now needed because we not only need a +5 volt and ground (return) line, but we need the unregulated line which provides between 7 and 11 volts- for use by the external 16K RAM.

4. Mount the 7805 regulator- with a good heat sink to the external power supply. Use silicone rubber to mount the heat-sinked 7805. Now rig 3 lines via a 3 wire plug and jack to provide variable DC (7-11 v),regulated +5VDC and a ground. Don't forget to put a SPST miniature switch on the external power supply.

```
                  7-11  V
            ____________ ____)
                             )
------        ------         
      )      (      )5 V Reg)
      )___/___(7805  )__ ____) Computer
ext.  )      (______)       )
power )                      )
supply)                      )
      )  ground              )
______)__________________ ____)
                             )_________
```

SINCUS is not responsible for the results of any changes that page 5

"YOUR 2068 STORE"

"YOUR 2068 STORE

KNIGHTED COMPUTERS

707 Highland St.
Fulton, N.Y. 13069

HARDWARE AND SOFTWARE FOR THE

TIMEX sinclair TS 2068

A&J MODEL 2000 MICRODRIVE

for your TS2068 - $199.50
Transfer Rate = 11,400 Baud !!!

INCLUDES:

Microdrive and Interface
5 Microwafer II (1 ea. size)
Expansion Cartridge
Wafer Organizer - Wafer Wheel
Wafer Caddy
Owners Man. & 90 Day Warranty

Eliminate those LOAD/SAVE problems with this truly great system that approaches the speed of many of the current Disk Drives -- at half the cost !!! This system does not use any of the RAM in your computer like many other of the mass storage devices available do!

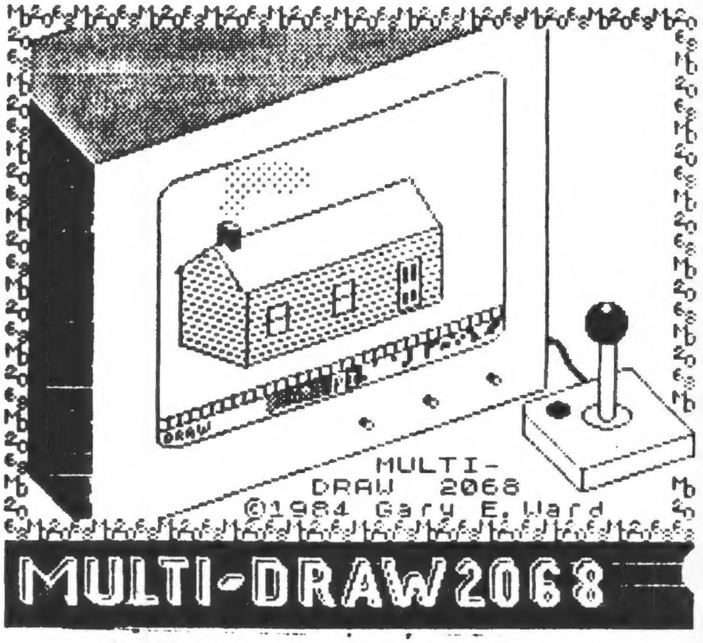
TASWORD II by Tasman Software

A very professional and economical 2068 word processor for the TS2068. Features automatic wordwrap, de-selectable right justification, block move, copy, insert, help displays, line centering, re-formating, find and replace. Comes with a very nicely written manual and a truly great "TASWORD TUTOR" on the tape to help you easily learn TASWORD II. Your choice of 64 or 32 CHARACTERS ON SCREEN !! Can be used with 2040 printer or AERCO or TASMAN I/F. ==> ==> $48.95

LEGEND 880 FULL SIZE PRINTER
Features the latest advances in dot matrix printing using NEW SQUARE DOT TECHNOLOGY.
Timex compatible. I/F req.

INTRODUCTORY PRICE......

$249.00



The finest drawing program we've seen for the TS2068! This program allows you to draw on the screen with only the use of the joystick - change colors with the joystick- define characters with the joystick - turn on or off any pixel(s) at will and output to a TS2040 printer OR a full size printer (AERCO OR TASMAN I/F). Magnify instantly. NOW ONLY $19.95 on Cassette///Avail on Wafer soon

---

PRO/FILE 2068 by Thomas Woods - - - - - -	$ 24.95
PENETRATOR (THE ORIGINAL FROM TIMEX)- - -	$ 17.95

TS2068 SOFTWARE ON CARTRIDGE

ANDROIDS -	$ 19.95
PINBALL --	$ 21.95
FLIGHT SIMULATOR -	$ 21.95

TS2068 CASSETTE SOFTWARE

STOCK MARKET SIMULATOR
(finally avail.) $ 17.95

*** WRITE FOR OUR FREE CATALOG *** MANY OTHER TS2068 ITEMS ***

Add $3.00 to total order for shipping and handling
Use mailing address above

< MTERM SMART II patch for AERCO I/F >

To install the patch to your program, load your MTERM SMART II PROGRAM,
POKE the following data into the locations listed and SAVE the program using
 ******* SAVE"patch"CODE 54016,7721 ********

```
POKE 59173 to 195   POKE 59174 to 126   POKE 59175 to 229
POKE 59188 to 195   POKE 59189 to 178   POKE 59190 to 229
POKE 58618 to 42    POKE 58619 to 232   POKE 58620 to 238   POKE 58621 to 237
POKE 58622 to 75    POKE 58623 to 236   POKE 58624 to 238   POKE 58625 to 3
POKE 58626 to 125   POKE 58627 to 185   POKE 58628 to 32    POKE 58629 to 7
POKE 58630 to 124   POKE 58631 to 184   POKE 58632 to 32    POKE 58633 to 3
POKE 58634 to 195   POKE 58635 to 80    POKE 58636 to 229   POKE 58637 to 62
POKE 58638 to 127   POKE 58639 to 219   POKE 58640 to 254   POKE 58641 to 31
POKE 58642 to 210   POKE 58643 to 80    POKE 58644 to 229   POKE 58645 to 219
POKE 58646 to 127   POKE 58647 to 230   POKE 58648 to 16    POKE 58649 to 32
POKE 58650 to 242   POKE 58651 to 126   POKE 58652 to 254   POKE 58653 to 10
POKE 58654 to 32    POKE 58655 to 2     POKE 58656 to 62    POKE 58657 to 0
POKE 58658 to 254   POKE 58659 to 12    POKE 58660 to 32    POKE 58661 to 2
POKE 58662 to 62    POKE 58663 to 0     POKE 58664 to 205   POKE 58665 to 48
POKE 58666 to 229   POKE 58667 to 35    POKE 58668 to 195   POKE 58669 to 2
POKE 58670 to 229   POKE 58671 to 0     POKE 58672 to 211   POKE 58673 to 127
POKE 58674 to 0     POKE 58675 to 219   POKE 58676 to 127   POKE 58677 to 201
POKE 58678 to 0     POKE 58679 to 0     POKE 58680 to 0     POKE 58681 to 0
POKE 58682 to 0     POKE 58683 to 0     POKE 58684 to 0     POKE 58685 to 0
POKE 58686 to 0     POKE 58687 to 0     POKE 58688 to 0     POKE 58689 to 0
POKE 58690 to 0     POKE 58691 to 0     POKE 58692 to 0     POKE 58693 to 0
POKE 58694 to 0     POKE 58695 to 0     POKE 58696 to 0     POKE 58697 to 0
POKE 58698 to 0     POKE 58699 to 0     POKE 58700 to 0     POKE 58701 to 0
POKE 58702 to 0     POKE 58703 to 0     POKE 58704 to 62    POKE 58705 to 13
POKE 58706 to 205   POKE 58707 to 48    POKE 58708 to 229   POKE 58709 to 62
POKE 58710 to 13    POKE 58711 to 205   POKE 58712 to 48    POKE 58713 to 229
POKE 58714 to 62    POKE 58715 to 13    POKE 58716 to 205   POKE 58717 to 48
POKE 58718 to 229   POKE 58719 to 33    POKE 58720 to 117   POKE 58721 to 239
POKE 58722 to 54    POKE 58723 to 13    POKE 58724 to 6     POKE 58725 to 32
POKE 58726 to 43    POKE 58727 to 54    POKE 58728 to 32    POKE 58729 to 16
POKE 58730 to 251   POKE 58731 to 125   POKE 58732 to 50    POKE 58733 to 231
POKE 58734 to 238   POKE 58735 to 201   POKE 58736 to 33    POKE 58737 to 195
POKE 58738 to 229   POKE 58739 to 126   POKE 58740 to 167   POKE 58741 to 32
POKE 58742 to 4     POKE 58743 to 62    POKE 58744 to 35    POKE 58745 to 119
POKE 58746 to 201   POKE 58747 to 175   POKE 58748 to 24    POKE 58749 to 251
POKE 58750 to 203   POKE 58751 to 79    POKE 58752 to 194   POKE 58753 to 40
POKE 58754 to 231   POKE 58755 to 197   POKE 58756 to 229   POKE 58757 to 42
POKE 58758 to 176   POKE 58759 to 229   POKE 58760 to 237   POKE 58761 to 75
POKE 58762 to 174   POKE 58763 to 229   POKE 58764 to 175   POKE 58765 to 237
POKE 58766 to 66    POKE 58767 to 32    POKE 58768 to 4     POKE 58769 to 175
POKE 58770 to 225   POKE 58771 to 193   POKE 58772 to 201   POKE 58773 to 219
POKE 58774 to 127   POKE 58775 to 230   POKE 58776 to 16    POKE 58777 to 32
POKE 58778 to 246   POKE 58779 to 10    POKE 58780 to 205   POKE 58781 to 48
POKE 58782 to 229   POKE 58783 to 3     POKE 58784 to 62    POKE 58785 to 96
POKE 58786 to 184   POKE 58787 to 32    POKE 58788 to 3     POKE 58789 to 1
POKE 58790 to 0     POKE 58791 to 94    POKE 58792 to 237   POKE 58793 to 67
POKE 58794 to 174   POKE 58795 to 229   POKE 58796 to 24    POKE 58797 to 227
POKE 58798 to 0     POKE 58799 to 94    POKE 58800 to 0     POKE 58801 to 94
POKE 58802 to 229   POKE 58803 to 219   POKE 58804 to 115   POKE 58805 to 254
POKE 58806 to 12    POKE 58807 to 40    POKE 58808 to 4     POKE 58809 to 254
POKE 58810 to 10    POKE 58811 to 32    POKE 58812 to 1     POKE 58813 to 175
POKE 58814 to 245   POKE 58815 to 42    POKE 58816 to 176   POKE 58817 to 229
POKE 58818 to 119   POKE 58819 to 35    POKE 58820 to 62    POKE 58821 to 96
POKE 58822 to 188   POKE 58823 to 32    POKE 58824 to 3     POKE 58825 to 33
POKE 58826 to 0     POKE 58827 to 94    POKE 58828 to 34    POKE 58829 to 176
POKE 58830 to 229   POKE 58831 to 241   POKE 58832 to 225   POKE 58833 to 201
```

To toggle the printer ON/OFF while receiving data use CAP SHIFT 8 and P(PRINT).
To print the BUFFER DATA use V and P options from the BUFFER MENU.
The BUFFER CON mode should be set to NONE. End printing with CAP SHIFT & BREAK.
LINE FEEDS and PAGE FEEDS are filtered. To enable the codes POKE 58659 & 58806 to 0 for PAGE FEED or 58653 & 58810 to 0 for LINE FEED.
Your comments appreciated. For questions, send stamped self addressed envelope.

Dave Schoenwetter
1335 Farm to Market Rd.
Endwell, NEW YORK, 13760
(607) 748-9687

COMPUTUS INTERRUPTUS

-OR, THE JOY OF USING INTERRUPTS ON YOUR COMPUTER

by Wes Brzozowski, SINCUS

Part one

"All right", comes the chorus, "what's an interrupt, and why should I care?" I'll admit, it's possible to lead a normal, happy life even if you've never heard of an interrupt. But in that case, you'll have missed something that's at least lots of fun and, at most very useful.

This series of articles will try to give something to everyone. Those who despise technical details will be able to pick out some programs that can be entered and immediately used, to give new power to their computers. Beginning machine code programmers will learn of a hidden "bug" in the system that can do weird things to their software. Advanced machine code programmers will find a versatile tool that will allow them to do things they may not have suspected possible. Those who like to build hardware will also find a few interesting tricks. By the end of this first article, we'll understand what interrupts are and have a small program that demonstrates "interrupts in action", and which may be of use, once incorporated into a BASIC program. We'll build on this demonstrator program in the future.

In order to accomodate the many levels of experience of various SINCUS members, this article is laid out in topics. Each starts with simple explanations and progresses into technical detail. If you find yourself in "too deep", the water becomes shallow again at the start of the next topic! I've never written in such a manner before, so comments on its degree of success (or failure!) will be much appreciated (or tolerated!). Seriously, any suggestions on how to present technical ideas to as wide an audience as ours will be very useful.

#1. WHAT'S AN INTERRUPT?

Perhaps an analogy would be the best way to begin. Suppose, while you're reading this article, the telephone rings. You'll probably set the newsletter down, mentally remembering where you were and go answer the phone When you're done, you'll come back and resume where you left off. You've just *"serviced an interrupt."* Let's try another analogy. Suppose you find my articles so interesting that you absolutely can't be disturbed while reading them. Because of this, you unplug your phone before you start reading and plug it back when you're done. If the world outside tries to interrupt you, you won't know and won't respond. During that time, you've *"disabled the interrupt."* Now for one more analogy. Your neighbor knows you have a habit of unplugging your phone, so he comes to your house and rings your doorbell. He can see you through the window so you can't ignore him. You set down the newsletter, and open the door. You are*"servicing a non-maskable interrupt."*

The TS 2068 has both maskable and non-maskable interrupts, activated by pulling one of two pins on the expansion connector to ground. When this happens, the present value of the program counter goes on the stack, and the machine starts executing at some new location, where the "interrupt handler" software is. If it will make it easier to picture, it acts as though a CALL (machine code, but very much like a GOSUB) instruction has been added right where the computer happens to be running code. In fact, the interrupt handler is written as a subroutine that actually can be CALLed. Exactly where in memory the interrupt handler may be located will be dealt with later.

The TS2068 generates its own (maskable) Interrupt every 1/60 second. This causes the keyboard to be scanned and the 3 byte system variable FRAMES to be increased by one count. This variable can be used as a clock or timer and, in fact is what the PAUSE instruction uses to determine whether it's waited long enough. (Have you noticed that the number that follows PAUSE is a count, *also* in sixtieth of a second?). This 60 Hertz interrupt is also synchronized to the beginning of each video frame on your TV or monitor, which can be useful. It's not hard to divert this interrupt so it can do some work for us on top of its normal duties. We'll demonstrate this in a moment.

#2 CAN'T AN INTERRUPT DISRUPT A PROGRAM THAT IS RUNNING?

Absolutely. One place where our "phone answering anaology breaks down is in the fact that you remember having answered the phone, but the routine being interrupted "has no knowledge" that it's been temporarily set aside. This means that the interrupt handler software has to be carefully written so as not to change *anything* unexpectedly. For example the first thing usually done is to PUSH all registers onto the stack. The last thing it does is to POP them all back into place before it RETurns to the program that was interrupted. Therefore, even though the interrupt handler may have temporarily changed the registers, it leaves them exactly as it "found" them.

#3.WHAT ABOUT PROGRAMS WHERE THE EXACT TIME REQUIRED TO EXCUTE A LOOP IS CRITICAL? WON'T AN INTERRUPT CHANGE THAT TIMING?

Yes, it would, in such circumstances, an interrupt could be disastrous. When such things are expected, (LOADing, SAVEing, BEEping, LPRINTing, are all examples) the maskable interrupt is disabled with the DI machine code instruction. The non-maskable interrupt cannot be disabled, and could be quite disruptive, if misused. It is normally not used with the TS2068, and a ROM bug generated by Sinclair and faithfully copied by TIMEX, makes it nearly impossible to use, any way. Next time, we'll investigate some hardware methods that get around this bug.

The following "experiments" show how things can go when unexpected interrupts appear, or when necessary interrupts fail to materialize. I've mentioned that the TS2068 generates its own maskable interrupt (from now on, we'll just call it "the interrupt") every sixtieth of a second. This can be turned off in hardware by setting bit 6 or I/O port FF. It's not quite the same as execut- ing a DI, but it has the same effect, and can be done from BASIC. TYPE IN:

```
10 OUT 255,64
20 PAUSE 5
```

If you RUN 20, the program runs in a flash; PAUSE 5 doesn't take very long, after all. However, if you just RUN, the computer is "locked up" until you shut off the machine. Line 10 shut off the interrupts. (The analogy now is not so much like unplugging your phone as it is shutting down the phone company! Fortunat recent actions by the U.S. Justice Department have prevented this analogy from seeming overly bizarre.) Remember the systems variable FRAMES is incremented every time an interrupt occurs. PAUSE 5 waits for it to get incremented 5 times. Unfortunately, with no interrupts, FRAMES doesn't change, and the computer sets out to prove that it's more patient than its owner!

For the case where we don't want interrupts, those who own or can get access to a TS2040 PRINTER may type in the following:

```
10 PRINT AT 10,10;"WES"
20 RANDOMIZE USR 2562
30 STOP
40 PRINT AT 10,10;"WES"
50 RANDOMIZE USR 2563
```

The ROM routine at 2563 contains the COPY command. If you RUN this, you'll get a piece of paper with my name on it. However, the 2040 printer is controlled by a precisely timed set of pulses. An interrupt would cause some of these pulses to "be lost". For this reason, the first instruction in the COPY command is DI, which disables the interrupt. If we instead RUN 40, we will have skipped around the DI instruction, and the print sequence is disrupted 60 times a second by unwanted interrupts. This time, my name comes out as a meaningless blur. I liked the first way better!

The moral to machine code programmers is, no matter how tight the little loops in your programs, the computer is sneaking in 60 times a second unless you DI first. Do that DI before entering any critical timing loops and restore things later with EI. Don't forget that the keyboard won't be scanned and FRAMES won't be updated while your DI is active.

#4 WHERE DOES THE INTERRUPT HANDLER HAVE TO BE PLACED IN MEMORY? CAN I PUT IT WHERE I WANT IT, OR ADD MY OWN HANDLER?

You have a little control, in some cases. The non-maskable interrupt always starts at location 0066H. In the TS2068, this is in the ROM. I mentioned a bug there that keeps us from normally using this feature. If any one has built my Universal AROS/LROS BOARD (SINCUS NEWS, Nov 84) and fitted it with RAM memory, , can simply load in Spectrum BASIC, change the bug, and go. We'll discuss this next time, along with a different hardware method to correct the "bad byte" using the TIMEX ROM.

The maskable interrupt operates in 3 software selectable modes. MODE 0 causes the interrupt to start executing at a location defined solely by external hardware. We won't use it here, but it's mentioned for completeness. MODE 1 causes the interrupt to start executing at location 0038H. This is how the TS2068 normally operates, and the interrupt handler is located there. The TS2068 Technical Manual in Section 5.3.1 suggests a totally worthless method of interecepting the MODE 1 interrupt;I consider it worthless because it can't be used along with BASIC. Let's be greedy and demand it all. Once again, we can use the AROS/LROS Board with a change to the interrupt handler, but this still requires one to build the board. Let's demand a software only BASIC-compatable technique. It turns out that one exists!

Our ability to easily use the interrupt lies in interrupt MODE 2. In it, the most significant byte of an address is kept in the Z-80's "I" register. The least significant byte is read from the databus. <u>IMPORTANT</u>: users of Spectrum Emulators should note that Real Spectrums put a different value on the data bus (FF) than do TS2068's (I've detected 0F, 2F, 3F and 0E so far, with evidence that there may be others). For this reason, certain Spectrum software that uses interrupt MODE 2 won't work on a TS2068, even with an emulator. It appears that ing pullup resistors in the data bus fixes this problem. (Only bit 2 already <u>has</u> a pullup resistor; probably used by code at location 0BFF in the EXROM, for detecting whether additional memory expansion banks are present. This will be the subject of another article, but it's worth pointing out here to explain why only 7, not 8, resistors are needed to enhance Spectrum emulators.)

In the spirit of true greediness, wanting our own interrupt handler to work even without pullup resistors, we will want to tolerate <u>any</u> value on the data bus. We even want to tolerate <u>variable</u> values on the data bus. Fortunately, there's a renegade Spectrum add-on joystick that does just such a thing during interrupts. This is fortunate for us, because it's caused our British friends to solve the problem for us.

One thing I haven't mentioned is that the address assembled from the "I" register and the data bus is <u>not</u> the address of the interrupt handler. Although this makes it a bit more difficult to understand MODE 2, it lets us put the handler wherever we want; we can even change it easily while a program is running

In designing our interrupt code, we'll borro rather heavily from the solution proposed by Tom Webb, in <u>Advanced Spectrum Machine Language</u>, Melbourne House, 6.95 pounds. If we put FE in the "I" register, but don't know what will appear on the data bus, the machine will get the address of the interrupt handler from somewhere between locations FE00 and FF00. If we fill this 257 byte block with FD's, then the address of the interrupt handler will always be FD FD! This is 3 bytes before the block of FD's, and is just long enough for a JP instruction to the real interrupt handler. Doing this, our "software only" fix for the hardware problem takes up only 260 bytes of memory, and it's all in one continuous block! We have 255 bytes of memory available above the FD block and it would be most convenient to locate our interrupt handler there. We'll end the handler with a JP to the ROM interrupt handler, so that the keyboard will still be scanned, as usual. (Being lazy as well as greedy, we'd rather not do that ourselves!)

#5 CAN WE DO SOMETHING USEFUL WITH THIS HANDLER?

There's nothing wrong with being practicle, so why not? There are a number of Spectrum programs that use MODE 2 to actually add new commands to BASIC. The following program will give a much simpler, but distinctly related, example by adding a new function to the TS2068. As long as the interrupt is enabled, you can immediately COPY the screen to the printer by simultaneously pressing SYMBOL SHIFT and BREAK. This can even be done while a program is running, and even in the middle of a PRINT statement. When the copy is done, the program will continue, completely oblivious to the fact that it's been interrupted. The printout will include the edit line.

Certain BASIC commands disable the interrupt. During such intervals, this copy-screen function won't work. These commands are (LOAD, SAVE, VERIFY, MERGE, COPY, LLIST, LPRINTN and BEEP). Some commercial machine code programs also disable the interrupt

Add the following to your own BASIC program (exact line numbers aren't important, as long as you get the lines in the right order.) Make sure your program executes it once; more times won't hurt, but they won't help, and take a few seconds to run. Once this is done, the copy-screen command is active, and will remain so, even if you STOP the program and LOAD in a new one. NEW shuts off the interrupt mode, but leaves the code intact, so that it can be reactivated with only the RANDOMIZE USR statement An example of some Transylvania Tower screens, taken "on the fly" are shown on ~~the next page~~-see page 3!!

We'll save a discussion of the program for next time, and we'll discuss the problems of relocating it, and how to modify it to print only part of the screen. Until then, the machine code group might get some enlightenment/amusement/frustration by "taking it apart", to see how it works. The correct answer will be printed here next time!

Wes Brzozowski
SINCUS

Of Sorts...on sorts...sort of...

by Tony Cekolin, SINCUS Corresponding Member
from Mobile, Alabama

When programming a computer to handle large amounts of data, it is helpful to be able to put that data into some orderly form and then be able to retrieve it quickly. That process is called sorting and searching and if you're like me, the whole process has been something of a mystery to you. How do you get a computer to put a data set, be it numeric or alphbetic, into order and how should you tell the computer to go look for it?

Sorting is the group of methods used to put that information into some logical order. I really don't know how many different types of sorting routines there are available, but there are a bunch. About the fastest sort I have come across outside of machine code routines is the Shell-Metzner sort. It works by looking at pairs of data and performing a swap when necessary. It finds the pairs to look at by logarithmic calculations and so I won't try to explain the method (especially because I can't -- you'll just have to take my word for it; it is fast). I will reproduce the code though.

```
1470 REM $ $ $ SORT $ $ $
1500 LET A = INT (LN N/LN 2):  LET F = 2^A-1
1510 LET F = INT (F/2):  IF F = 0 THEN RETURN
1520 LET D =N-F: LET B =1
    ( HERE N IS THE DIMENSION OF THE ARRAY  A$  AS
      THIS VERSION IS FOR AN ALPHABETICAL SORT)
1530 LET A = B
1540 LET E = A + F: IF A$(A)<A$(E) THEN GOTO 1570
1550 LET B = B + 1: IF B>D THEN GOTO 1510
1560 GOTO 1530
1570 LET T$ = A$(A): LET A$(A) = A$(E)
1575 LET A$(E) = T$
 ( T$ IS A TEMPORARY STRING USED TO PERFORM THE SWAP)
1580 LET A = A - F: IF A<1 THEN GOTO 1550
1590 GOTO 1540
```

<u>To test this module, enter as direct commands:</u>

```
DIM A$(5)
LET A$(1)="B"
LET A$(2)="E"
LET A$(3)="A"
LET A$(4)="D"
LET A$(5)="C"
LET N=5:GOTO 1470
```

The way the module stands, the letters should be sorted in reverse order. To get them sorted in the correct order use ">" instead of "<" in line 1540. This module can be used with numeric data by using a numeric array with "N" items.

This sort is very close to the famous bubble sort in the way it operates except that the Shell-Metzner sort is faster. The Bubble sort is based on a swap in the same manner except the swaps are done sequentially instead of logaritmically.

To use a Bubble sort you would have an array dimensioned with N items and then use a loop to compare each item with the next one, swapping when necessary.

```
FOR I = 1 TO N-1
FOR J = 1 TO N-1
IF A$(I)>A$(I+1) THEN GOTO 2ND LINE DOWN
NEXT J
LET T$=A$(I)
LET A$(I)=A$(I+1)
LET A$(I+1)=A$(I)
NEXT J:NEXT I
```

Again T$ is just a temporary string to store the first string in while you swap. The Bubble sort can be used with numeric data just as easily.

All the sorts available can be modified to be even faster when you are working with large strings of information. To do this you create a seperate array and fill it with the numbers of the dimension of the string array. For instance, if you have a string array with 100 large items in it you fill the numeric array with the numbers 1 to 100 in order. Then you use the numbers to refer to the location of the information in the string array. So your comparison statement becomes:

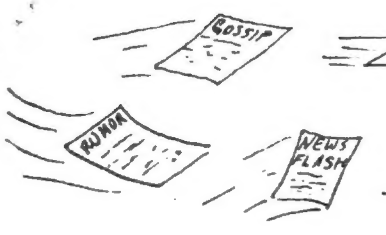
```
IF S$(A(I))>S$(A(I+1)) THEN  GOTO  SECOND LINE DOWN
NEXT I
```

<u>Then sort the numeric array:</u>

```
LET T=A(I)
LET A(I)=A(I+1)
LET A(I+1)=A(I)
NEXT I
```

What you end up doing is to sort the reference to your string array which is much faster than moving large blocks of information. This method is called tag sorting because you tag each item in your string array with a number and then move the number around. Just as there are many kinds of sorts, there are also many different kinds of searches. Any time you want to get a particular piece of information out of your database you have to locate it first. The simplest way to do that is to look at each item until you come up with a match. This is known as a sequential search. That is fine for small amounts of information to search through but if the number of items ig fairly large you can be waiting quite awhile. The number of comparisons can be found by dividing the number of items to be searched by two. That is if you have 55,000 items, your routine will have to look at 27,000 of them on the average. Not good! The code for the sequential search is:

```
FOR I = 1 TO THE NUMBER OF ITEMS
IF A$(I)= TARGET THEN  GOTO 2ND LINE
NEXT I
LET T$=TARGET
LET I=NUMBER OF ITEMS
NEXT I
```



2K-EXPRESS

EDIT - some changes in SINCUS NEWS, starting with new dues for corresponding members, $8/year. This is the result of the new postal rate of $.22 per newsletter plus we are publishing 12 per year as compared to the 10 issues we originally commited for. I trust this nominal increase will be accepted by all conserned, especially in view of the improved quality and prompt mailing of SINCUS NEWS!

NEW - this issue is the first that is being "printed" locally. My "test" copies indicate that 64 characters per column is legible, so I am using that format throughout.

Of Sorts...on sorts...sort of... (continued)

By letting I equal the number of items you are able to jump out of the loop, or you could use a goto to jump out.

There is a faster searching method called the Binary search. It starts by looking for the middle of your data set and then asking if your target is greater than, less than, or equal to the middle item. If the target is greater than the middle then the search throws out (figuratively -- not really!!) all the items that are smaller than the target. It then finds the middle of the remainder and asks again. The same thing happens in reverse if it is less than the middle. When the target is equal to the item looked at then you are done. If you have a data set of one million items, search has to only make 20 comparisons to find the target.

Convinced? Well the proof is tough but that is what I'm told. The code runs like this:

```
                510 LET L = 1
                520 LET H = N
                530 LET C = 0
                540 LET M = INT((H+L)/2)
                550 LET C = C + 1
                560 IF X = A(M) THEN GOTO (JUMP OUT OF LOOP)
                570 IF L>= H THEN JUMP OUT BECAUSE THE ITEM IS
                   NOT TO BE FOUND
                580 IF X>A(M) THEN GOTO 610
                590 LET H = M -1
                600 GOTO 540
                610 LET L = M + 1
                620 GOTO 540
```

In line 520 N is the number of items to be searched.

For more information on sorts and searches I strongly recommend a look at *THE ESSENTIAL GUIDE TO TIMEX/SINCLAIR HOME COMPUTERS* especially the section between pages 292-318. The authors provide a very strong discussion on the subject.

Tony Cekolin
SINCUS

PEEKing Ahead - the 2050 modem software has been modified by Dave Schoenwetter so that you can use a full sized printer, the incoming signal can go right to the printer or it can go into the buffer and then you can send it to be printed out. This is the product of many hours of hard work and trial and error-we do recognize Dave for this effort and I hope all"telecommunicators" will take the time to modify their "original copy of Mterm using the POKEs found on page 6. Thanks Dave for the really big effort!

GOTO the meetings!! We are running classes in programming and are starting a new series of meets on "hardware" and "TS1000" (each is separate!) in addition to the monthly meeting. So come on down and meet some neat folks and stretch your head a little!

LPRINT - or something like it is available on "Vu-File"-look for your newsletter to have a peal-off type label next month!!! And it was so SO simple-just throw a switch on the printer!!!

RUN to the nearest computer and get Wes' program to work-you will learn something!!!

EDITOR's NOTE - TASWORD TWO and the AERCO Interface

I have seen a couple "printer patches" for the AERCO I/F and TASWORD TWO. The following is what I was given and have been using for the past six months. Thanks to Knighted Computer for this information!!!

Load TASWORD TWO w/o the AERCO "software" then select "b" for BASIC and then use the POKE command to make the following POKEs:

```
POKE 57578,32
POKE 57579,12

POKE 57999,127
POKE 58000,230
POKE 58001,19
POKE 58002,254
POKE 58003,1
POKE 58004,32
POKE 58005,-8
POKE 58006,241
POKE 58007,211
POKE 58008,127
POKE 58009,0
POKE 58010,219
POKE 58011,127
POKE 58012,201
```

Now PEEK each of the above addresses to confirm the correct entry of the decimal values. If any are wrong "POKE" it over! Now return to the program - use GOTO 1 and the "STOP" to get the main menu. Now select "t" to save TASWORD-label the cassette!!!